LIBRARY OF FUNCTIONS FOR CONTACT DETECTION WITH APPLICATIONS IN BIOMECHANICS

FINAL PRESENTATION

LEBiom's Final Project

Supervised by Prof. Daniel Simões Lopes

TABLE OF CONTENTS

MOTIVATION

VARIETY OF FIELDS

Collision Detection can be used in: astronomy, physics, molecular geometry, electromagnetics, fluid mechanics…

BIOMEDICAL RELEVANCE

Simulating foot-ground contact[1] & virtual reality-based training system for needle micromanipulation[2].

MULTIPLE VIRTUAL APPLICATIONS

Virtual reality training, video games, rapid digital prototyping, and robotics simulation make use of **Collision Detection** and **Shortest Distance calculations**.

[1] “A superellipsoid-plane model for simulating footground contact during human gait” (Lopes et al., 2015)
[2] “Multisensory learning cues using analytical collision detection between needle and a tube” (Wang et al., 2004)

CHALLENGE

Collision detection has attracted the attention of researchers for decades in the field of computer graphics, robot motion planning, computer-aided design...

&

A large number of successful algorithms have been proposed and applied.

BUT

! **A compilation of these solutions is lacking.**

AND

There's a need to increase the computational performance of already existing algorithms.

GOAL

What we proposed to do:

1. Review the literature.
2. Compile the functions into a systematized library.
3. Create a game in Unity to apply and interact with the library created.
4. Develop a small Biomechanical simulation.

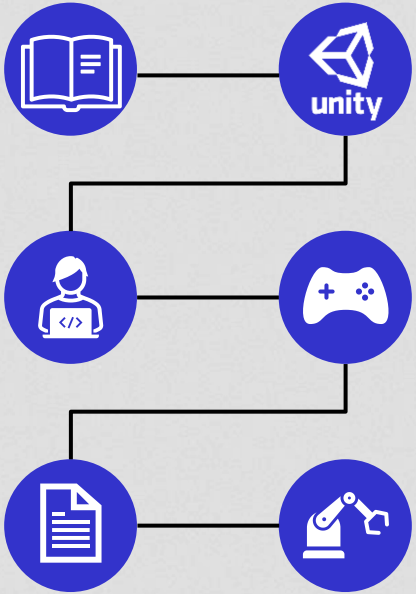
STRATEGY

STEP 1

Literature review.

STEP 2

Get familiar with Unity and C# scripting.



STEP 3

Development of a library of functions.

STEP 4

Development of a Unity game implementing the created library.

STEP 5

Publish the code and write down a paper on "state-of-the-art“.

STEP 6

Implementation of case studies in movement biomechanics.

LITERATURE REVIEW

30 references

- **5** books
- **20** papers
- **5** online repositories

CLOSEST DISTANCE TABLE 2D (26 contact pairs)

Contact Pairs	References	Used References	Specific Location/Equation
2D			
Point-Line	Scheneider, J.P., Eberly, D. H., "Distance in 2D", "Point to Linear Component", "Point to Line". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 190-191.		
Point-Ray	Scheneider, J.P., Eberly, D. H., "Distance in 2D", "Point to Linear Component", "Point to Ray". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 191-192.		
Point-Segment	1. Scheneider, J.P., Eberly, D. H., "Distance in 2D", "Point to Linear Component", "Point to Segment". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 192-193. 2. Ericson C., "Basic Primitive Tests", "Closest-point Computations", "Closest Point on Line Segment to Point ". Real-Time Collision Detection. Elsevier Inc., 2005. p. 127-129		
Point-Polyline	Scheneider, J.P., Eberly, D. H., "Distance in 2D", "Point to Polyline". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 194-		

Closest Distance 2D table

(**26** contact pairs)

Closest Distance 3D table

(**36** contact pairs)

CLOSEST DISTANCE TABLE 3D (36 contact pairs)

Contact Pairs	References
3D	
Point-Linear Component	Scheneider, J.P., Eberly, D. H., "Distance in 3D", "Point to Linear Component". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 365-367
Point-Line-Segment	Scheneider, J.P., Eberly, D. H., "Distance in 3D", "Point to Linear Component", "Point to Ray or Line Segment". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 367-369.
Point-Ray	Scheneider, J.P., Eberly, D. H., "Distance in 3D", "Point to Linear Component", "Point to Ray or Line Segment". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 367-369.
Point-Polyline	Scheneider, J.P., Eberly, D. H., "Distance in 3D", "Point to Linear Component", "Point to Polyline". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 369-374.
Point-Plane	1. Scheneider, J.P., Eberly, D. H., "Distance in 3D", "Point to Planar Component", "Point to Plane". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 374-376. 2. Christer Ericson, "Basic Primitive Tests", "Closest-point Computations", "Closest Point on Plane to Point ". Real-Time Collision Detection. Elsevier Inc., 2005. p. 126-127
	1. Scheneider, J.P., Eberly, D. H., "Distance in 3D", "Point to Planar Component", "Point to Triangle". Geometric Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 376-382

Link dos vídeos: https://drive.google.com/drive/folders/1qskfVW6yp6mhC9OsbgQmolKIkwbyKSa3?usp=sharing

Collision Table 2D: (61 contact pairs)

Contact Pairs	References	Used Reference(s)	Specific Locatio Equation
2D			
Point-Point	1. Schwarzl, T. "Collision Detection: Point-Point Collision". 2D Game Collision Detection: An introduction to clashing geometry in games. CreateSpace Independent Publishing Platform, 2012. p. 29 2. jeffThompson, "CollisionDetection/CodeExamples/PointPoint at master · jeffThompson/CollisionDetection," GitHub, Dec. 12, 2018		
Point-Line	1. Schwarzl, T. "Collision Detection: Point-Line Collision". 2D Game Collision Detection: An introduction to clashing geometry in games. CreateSpace Independent Publishing Platform, 2012. p. 50 2. jeffThompson, "CollisionDetection/CodeExamples/LinePoint at master · jeffThompson/CollisionDetection," GitHub, Dec. 12, 2018		
Point-Line-Segment	Schwarzl, T. "Collision Detection: Point-Line-Segment Collision". 2D Game Collision Detection: An introduction to clashing geometry in games. CreateSpace Independent Publishing Platform, 2012. p. 51		

Collision Detection 3D table

(**89** contact pairs)

COLLISION TABLE 3D (89 contact pairs)

Contact Pairs	References
3D	
Line-Triangle	Scheneider, J.P., Eberly, D. H., "Intersection in 3D", "Linear Components and Planar Components", "Linear Components and Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 485-488.
Line-Polygon	Scheneider, J.P., Eberly, D. H., "Intersection in 3D", "Linear Components and Planar Components", "Linear Components and Tools for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 488-491.
Line-Disk	Scheneider, J.P., Eberly, D. H., "Intersection in 3D", "Linear Components and Planar Components", "Linear Component and for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 491-493.
Line-Polyhedra	Scheneider, J.P., Eberly, D. H., "Intersection in 3D", "Linear Components and Polyhedra". Geometric Tools for Computer Gra Kaufamann Publishers. 2003. p 493-498.
Line-Quadric Surface	Scheneider, J.P., Eberly, D. H., "Intersection in 3D", "Linear Components and Quadric Surfaces", "General Quadric Surfaces". Computer Graphics. Morgan Kaufamann Publishers. 2003. p 499-501.
Line-Sphere	Scheneider, J.P., Eberly, D. H., "Intersection in 3D", "Linear Components and Quadric Surfaces", "Linear Components and Sp for Computer Graphics. Morgan Kaufamann Publishers. 2003. p 501-503.
Line-Ellipsoid	Scheneider, J.P., Eberly, D. H., "Intersection in 3D", "Linear Components and Quadric Surfaces", "Linear Components and an

Collision Detection 2D table

(**61** contact pairs)

Link dos vídeos: https://drive.google.com/drive/folders/1I9xZX9wEYRHl_SiFW3PhfKdwdP6w5jJW?usp=sharing

CHOSEN PRIMITIVES

SHORTEST DISTANCE

Primitives in two and three dimensions:

- Point-Point;
- Point-Line Segment;
- Line Segment-Line Segment;
- Point-Circle/Sphere;
- Line Segment-Circle/Sphere;
- Circle/Sphere-Circle/Sphere.

COLLISION DETECTION

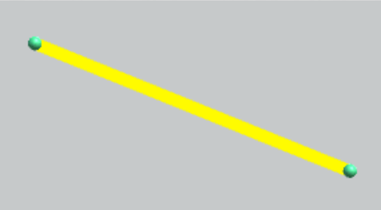
Primitives in two and three dimensions:

- AABB-AABB;
- OBB-OBB.

Primitives in three dimensions:

- Point-AABB;
- Point-Sphere;
- Sphere-AABB;
- Sphere-Sphere.

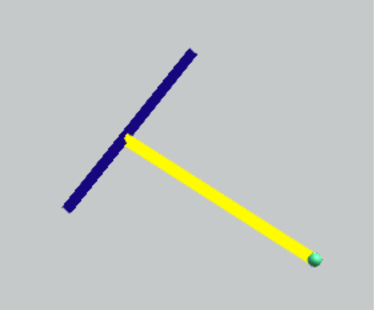
POINT-POINT



Given two points $\mathbf{P_1}$ and $\mathbf{P_2}$,

$$\boldsymbol{Closest\ Distance}: \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

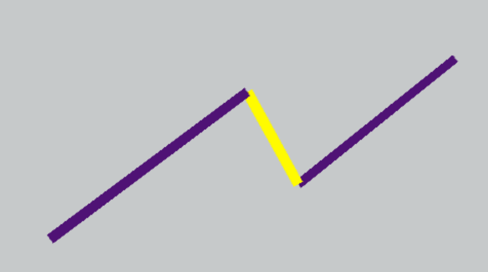
POINT-LINE SEGMENT



Given a point **Y** and a line segment $\boldsymbol{L(t)} = P + t\vec{v}$,

$$\boldsymbol{Closest\ Distance}: \begin{cases} \|Y - P\|, & if\ t' \leq 0 \\ \left\|Y - (P + t'\overrightarrow{v})\right\|, & if\ 0 < t' < 1 \\ \left\|Y - (P + \overrightarrow{v})\right\|, & if\ t' \geq 1 \end{cases}$$

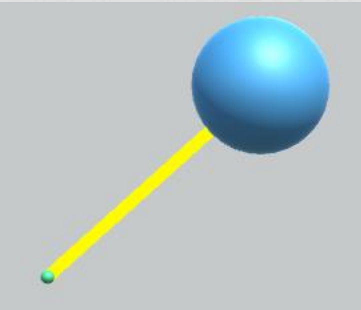
LINE SEGMENT-LINE SEGMENT



Given two parallel line segments $\boldsymbol{L_i(t)} = P_i + t\vec{d}_i$ $(i = 0{,}1,$ andt $t \in [0, T_i])$ and $\vec{\Delta} = P_0 - P_1$,

$$\textbf{\textit{Closest Distance}}: \begin{cases} \|\vec{\Delta}\|, \vec{d}_0.\vec{d}_1 < 0 \ \wedge \vec{d}_0.\vec{\Delta} \geq 0 \\ \left\|\vec{\Delta} + T_0\vec{d}_0\right\|, \vec{d}_0.\vec{d}_1 < 0 \ \wedge \vec{d}_0.(\vec{\Delta} + T_0\vec{d}_0) \geq 0 \\ \left\|\vec{\Delta} - T_1\vec{d}_1\right\|, \vec{d}_0.\vec{d}_1 < 0 \ \wedge \vec{d}_0.(\vec{\Delta} - T_1\vec{d}_1) \geq 0 \\ \ldots * \end{cases}$$

POINT-CIRCLE/SPHERE



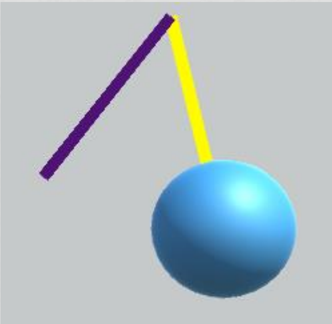
Given a points $\mathbf{P_1}$, the sphere's center $\mathbf{P_2}$ and the sphere's radius $\mathbf{r_1}$,

$$\textbf{\textit{Closest Distance}}: \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} - r_1$$

$$\textbf{\textit{Proximity Query}}: \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \leq r_1$$

*Too many cases to display.

LINE SEGMENT-CIRCLE/SPHERE



Given the sphere's center **Y**, the sphere's radius **r** and a line segment $\boldsymbol{L(t)} = P + t\vec{v}$,

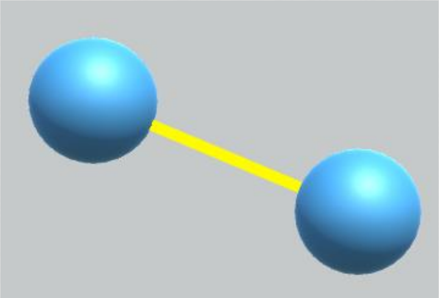
$$\boldsymbol{Closest\ Distance}: \begin{cases} \|Y - P\| - r, & if\ t' \leq 0 \\ \left\|Y - (P + t'\overrightarrow{v})\right\| - r, & if\ 0 < t' < 1 \\ \left\|Y - (P + \overrightarrow{v})\right\| - r, & if\ t' \geq 1 \end{cases}$$

CIRCLE/SPHERE-CIRCLE/SPHERE

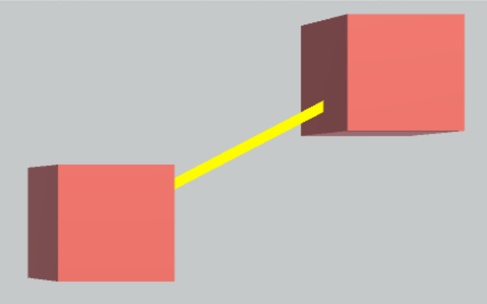
Given the sphere's centers $\mathbf{P_1}$ and $\mathbf{P_2}$ and the spheres' radii $\mathbf{r_1}$ and $\mathbf{r_2}$,

$$\boldsymbol{Closest\ Distance}: \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} - r_1 - r_2$$

$$\boldsymbol{Proximity\ Query}: \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \leq (r_1 + r_2)$$



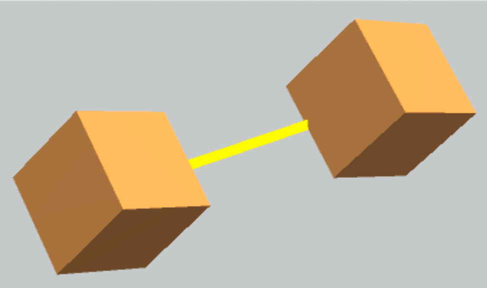
AABB-AABB



Given two AABBs **A** and **B**,

$$\textbf{Proximity Query}: \begin{cases} (A_{min_x} \le B_{max_x}) \wedge (B_{min_x} \le A_{max_x}) \\ (A_{min_y} \le B_{max_y}) \wedge (B_{min_y} \le A_{max_y}) \\ (A_{min_z} \le B_{max_z}) \wedge (B_{min_z} \le A_{max_z}) \end{cases}$$

OBB-OBB

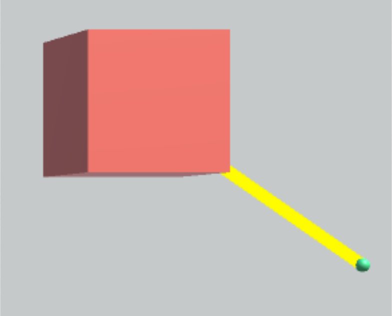


Given two OBBs, **A** and **B**, the vector joining their centers $\boldsymbol{d_{AB}}$, the boxes' extents $\mathbf{h_A}$ and $\mathbf{h_B}$ and a chosen separation axis, $\boldsymbol{L}$. Then,

$t = \|d_{AB} \cdot L\| - (\|h_A \cdot L\| + \|h_B \cdot L\|)$,

$$\textbf{Proximity Query}: \begin{cases} \exists L, t < 0 \Rightarrow No\ collision \\ \forall L, t \ge 0 \Rightarrow Collision \end{cases}$$

POINT-AABB



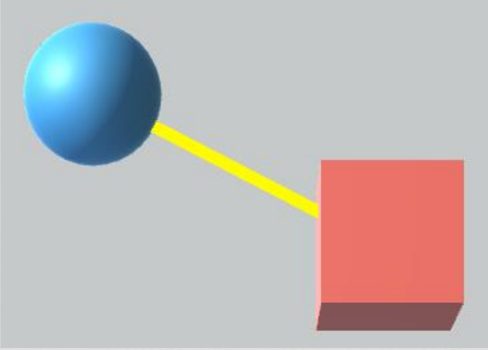
Given an AABB **A** and a point **P**,

$$\textbf{\textit{Proximity Query}}: \begin{cases} \left(P_x \le A_{max_x}\right) \wedge \left(A_{min_x} \le P_x\right) \\ \left(P_y \le A_{max_y}\right) \wedge \left(A_{min_y} \le P_y\right) \\ \left(P_z \le A_{max_z}\right) \wedge \left(A_{min_z} \le P_z\right) \end{cases}$$

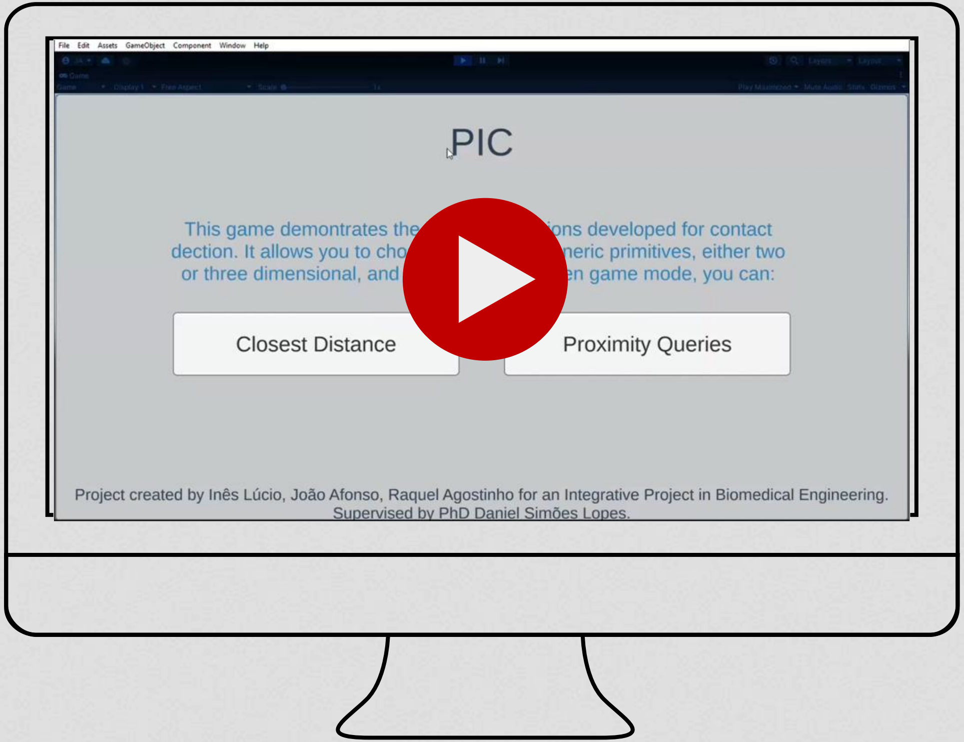
AABB-SPHERE

Given the AABB's closest point $\mathbf{P_1}$, the sphere's center $\mathbf{P_2}$ and sphere's radius $\mathbf{r_1}$,

$$\textbf{\textit{Proximity Query}}: \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \le r_1$$



UNITY MICROGRAME

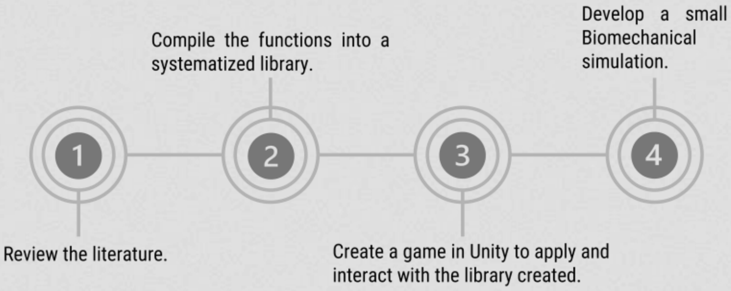


Link do vídeo: https://drive.google.com/file/d/1I-nE8S6uGAVRq8b31mflfXyv-chxXtKr/view?usp=sharing

ROBOTIC FOOT ON GROUND SIMULATION

Link do vídeo: https://drive.google.com/file/d/1sr5nXxCLY2mzsjYVgWBs8syQ7CCJ48qp/view?usp=sharing

FINAL REMARKS



All four of the goals completed

Work strategy followed through with the exeption of Step 5

FINAL REMARKS

Anyone can use the libraries developed to perform calculations of their interest !

OTHER POSSIBLE SIMULATIONS

- Hand-object contact for rehabilitation applications;
- Handrails for sports;
- Biomechanical seat design evaluation;
- Drug packaging.

IN THE FUTURE

- Add more complex primitives;
- Scale the libraries to other programming languages;
- Publish our work and share it with the community.

PROJECT MEMBERS:

FOR MORE QUESTIONS PLEASE CONTACT US:

Inês Lúcio – ines.marques.lucio@tecnico.ulisboa.pt
João Afonso – joaoduarteafonso@tecnico.ulisboa.pt
Raquel Agostinho – raquelmslagostinho@tecnico.ulisboa.pt

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.